

**UNITED STATES PATENT APPLICATION**

**of**

**Doug Durham**

**John Arrington**

**and**

**Dominic Coupal**

**for**

**SYSTEMS AND METHODS FOR  
CREATION AND USE OF  
A VIRTUAL PROTOCOL ANALYZER**

**WORKMAN NYDEGGER**  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

**SYSTEMS AND METHODS FOR  
CREATION AND USE OF  
A VIRTUAL PROTOCOL ANALYZER**

**BACKGROUND OF THE INVENTION**

**Related Applications**

[0001] This application claims the benefit of United States Provisional Patent Application Serial No. 60/442,277, entitled *DETECTION AND USER SELECTION OF PROTOCOL HARDWARE TYPES* filed on January 24, 2003, and incorporated herein in its entirety by this reference.

**The Field of the Invention**

[0002] The present invention relates to the identification, capture and analysis of data transmitted over a communications system. More specifically, embodiments of the present invention are concerned with the definition and use of a virtual protocol analyzer.

**Related Technology**

[0003] Many data communications systems use a variety of different data transmission mechanisms to enable communication between and among associated subsystems. In general, the type of data transmission mechanism employed in a given situation is determined with reference to the particular tasks desired to be accomplished in connection with that transmission mechanism and associated systems. Each different transmission mechanism, in turn, is associated with a particular transmission, or

WORKMAN NYDEGGER  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

communication, protocol that defines various parameters concerning the transmission of data in connection with the transmission mechanism. Such communication protocols commonly specify, for example, the manner in which data is encoded onto a transmission signal, the particular physical transmission media to be used with the transmission mechanism, link layers and other attributes.

[0004] As suggested above, a single data communications system may use multiple different transmission mechanisms and protocols. As an example, an enterprise may employ a communications system that uses five different data communication protocols, each adapted for a particular situation, wherein the five protocols may include: a first protocol for a high speed, inexpensive short-haul connection on the computer motherboard; a second high-bandwidth protocol for data center transmissions; a third protocol that is suited for efficiently transmitting information across the enterprise local area network ("LAN"); a fourth protocol adapted for high bandwidth, long haul applications; and, finally, a fifth communication protocol suited for data transmission to high performance disk drive storage systems. Thus, the typical communications system comprises a patchwork of different subsystems and associated communications protocols.

[0005] In this way, a communications system can be created that makes maximum and efficient use of the functionalities and capabilities associated with various different communications protocols. Further, advances in communications technology, coupled with declining costs, enable such communications systems to be implemented in a relatively cost effective fashion.

[0006] While communications systems that include components, devices and subsystems of varying protocols are able to exploit the respective strengths and useful features associated with each of the protocols, such multiple protocol systems can be problematic in practice. This is especially true where problem identification, analysis and resolution are concerned. In particular, the use of multiple communications protocols within the bounds of a single communications system greatly complicates the performance of such processes.

[0007] For example, as network data moves from a point of origin to a destination, by way of communication links, or simply “links,” the data passes through a variety of devices collectively representing multiple protocols. Typically, each such device modifies the network data so that the data can be transmitted by way of a particular link. However, modification of the data in this way often causes errors or other problems with the data. Such errors may occur as the result of various other processes and conditions as well. Thus, the various communication links in a communications system are particularly prone to introduce, or contribute to the introduction of, data errors. Moreover, data errors and other problems present at one location in the data stream may cause errors or other problems to occur at other locations in the data stream and/or at other points in the communications system and associated links.

[0008] One approach to problem identification, analysis and resolution in communications systems involves capturing a portion of the network data traffic. The captured data can then be retrieved for review and analysis. In some cases, such data capture is performed in connection with a protocol analyzer that includes various hardware and software elements configured and arranged to capture data from one or

more communications links in the communications system, and to present the captured data by way of a graphical user interface.

[0009] Generally, such protocol analyzers, or simply “analyzers,” capture data traffic in the communications system over a defined period of time, or in connection with the occurrence of predefined events. Use of the analyzer thus allows a network administrator to track the progress of selected data as that data moves across the various links in the communications system. Corrupted or altered data can then be identified and traced to the problem link(s), or other parts of the communications system. As discussed below, such protocol analyzers can provide useful results, but it is often the case that employment of typical protocol analyzers imposes unacceptable costs in terms of communications system performance and down time.

[0010] A typical protocol analyzer includes two or more ports configured and arranged to capture data on a communications link of a communications system. For high speed serial networks, two ports are needed to capture data on a bi-directional communications link. The two ports are sometimes referred to as a “port pair.” Typically, the user connects the ports in-line with the communications link and then sets up triggering criteria to be used as a basis for the capture of data passing through the communications link.

[0011] In some cases, protocol analyzers may have as many as 64 ports capable of capturing data, and the protocol analyzer may be capable of capturing data conforming with any of a variety of communications protocols. Typically, the ports of protocol analyzers have a native communications protocol, which corresponds to the data that the analyzer port is designed to monitor and capture. Examples of such protocols are

Gigabit Ethernet, Fibre Channel, and Infiniband. The ports in such protocol analyzers can be configured in a variety of ways. For example, the ports in the aforementioned 64 port protocol analyzer can be combined to create a single instrument having 32 port pairs. Alternatively, the ports can be configured into 32 different port pair instruments for use by 32 different users.

[0012] A significant problem with typical protocol analyzers however, is that if a network administrator or other user wishes to instrument a communications link for data capture and/or evaluation, the communications link must be broken and reconnected with the protocol analyzer. This is problematic at least because the process of breaking links significantly disrupts the operation of the communications system and increases communications system down time. The fact that users of conventional protocol analyzers are often required to break multiple communication links each time a different set of links is to be monitored further aggravates this problem.

[0013] In view of the foregoing, and other, problems in the art, what is needed are systems and methods for enabling a user to define and implement multiple virtual protocol analyzers, each of which may include a different combination of ports, within a single device. Such systems and methods should also allow the user to readily reconfigure a virtual protocol analyzer. Finally, exemplary embodiments of the invention should contribute to a relative reduction in the need to break communication links once the communication links have been initially instrumented.

**BRIEF SUMMARY OF AN EXEMPLARY EMBODIMENT  
OF THE INVENTION**

[0014] Generally, embodiments of the present invention are concerned with systems and methods for definition and use of virtual protocol analyzers in a communications system. In one exemplary implementation, a graphical user interface is provided for use in connection with a multi-protocol communications analyzer and enables a user to create a new domain, or modify an existing domain, that includes a list of ports selected by the user, where each port is associated with a particular communication link of the communications system. As used herein, “domain” refers to a combination of port pairs that collectively define the scope of application of a particular virtual instrument.

[0015] In operation, a user identifies the name of a domain that is to be created or modified for use in connection with the communications system. If a new domain is to be created, for example, the user then selects one or more ports from a list of available ports displayed by the graphical user interface. The selected ports are then associated with the new domain. Upon selection of the ports, the user can then set parameters for one, some or all of the ports according to various predefined options presented in connection with the graphical user interface. After the domain has been defined and the ports of the domain configured, the domain can then be used to analyze one or more of the links associated with the ports included in the domain.

[0016] In this way, multiple users are able to use a protocol analyzer and associated graphical user interface to define a variety of customized domains, employing different combinations of ports, configured to enable evaluation and resolution of particular problems. Further, once the communication links have been initially instrumented, the

WORKMAN NYDEGGER  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

definition and implementation of such domains can be accomplished without necessitating frequent physical disruption of the communication links in the communications system under test. These, and other, aspects of exemplary embodiments of the invention will become more fully apparent from the following description and appended claims.

WORKMAN NYDEGGER  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111



## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0017] In order that the manner in which the advantages and features of the invention are obtained, a particular description of the invention will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only exemplary embodiments of the invention and are not, therefore intended to be considered limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0018] Figure 1 is a schematic diagram that illustrates aspects of an exemplary data communications system that employs a variety of different data transmission mechanisms and protocols;

[0019] Figure 2 is a high level schematic diagram of a protocol analyzer such as may be employed in connection with a communications system;

[0020] Figure 3 is a schematic diagram of an exemplary multi-link protocol analyzer;

[0021] Figure 4 is a schematic diagram illustrating relationships between a protocol analyzer and various data transmission mechanisms employed in an exemplary data communications system;

[0022] Figure 5A is a flow diagram of an exemplary process for defining a virtual protocol analyzer;

[0023] Figure 5B illustrates an exemplary graphical user interface that displays a list of ports that have been discovered in a protocol analyzer and that is configured to enable a user to reserve a set of ports and create a domain representing a virtual protocol analyzer; and

[0024] Figure 5C illustrates an exemplary graphical user interface configured to enable a user to define the operating parameters of the ports in a domain.

WORKMAN NYDEGGER  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

## **DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE INVENTION**

[0025] Generally, embodiments of the present invention are concerned with systems and methods for definition and use of virtual protocol analyzers in a communications system. Such virtual protocol analyzers may also referred to herein as a “domain.” As disclosed herein, the user of the protocol analyzer is presented with a graphical user interface that enables the user to modify an existing domain or define a new domain by selecting from a list of available ports of a protocol analyzer. The domain thus encompasses a set of communication links from which data can be captured for analysis.

[0026] In this manner, the network administrator or other users can readily define and modify any number of domains for use in network traffic capture and analysis. Moreover, the definition and use of such domains can be accomplished with minimal disruption to communications system operations.

### **I. Exemplary Operating Environment**

[0027] With attention now to Figure 1, details are provided concerning an exemplary operating environment wherein the systems and methods disclosed herein may be employed. In the illustrated arrangement, the operating environment takes the form of a communications system 100 wherein data is transferred between a central processing unit (“CPU”) of a computing device and a redundant array of independent disks (“RAID”) system. The illustrated communications system 100 is an exemplary operating environment only however and the systems and methods disclosed herein

WORKMAN NYDEGGER  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111

may, more generally, be employed in any other operating environment(s) where such functionality may prove useful.

[0028] In the illustrated arrangement, the communications system 100 includes a CPU 102 of a computing device (not shown) configured and arranged for serial communication with an Infiniband adapter 104, an Infiniband/GigE bridge 106, a GigE/synchronous optical network (“SONET”) bridge 108, a SONET/Fibre Channel bridge 110, and a RAID drive storage tower 112. Serial connections between these components are provided by a series of communications links. In particular, the CPU 102 and Infiniband adapter 104 are connected by a peripheral component interconnect (“PCI”) Express link 114. Downstream of the Infiniband adapter 104, an Infiniband link 116 connects the Infiniband adapter 104 with the Infiniband/GigE bridge 106. In similar fashion, a GigE link 118 connects the Infiniband/GigE bridge 106 with the GigE/SONET bridge 108, while the SONET link 120 connects the GigE/SONET bridge 108 with the SONET/Fibre Channel bridge 110. Finally, a Fibre Channel link 122 connects the SONET/Fibre Channel bridge 110 with the RAID drive storage tower 112.

[0029] Each of the aforementioned links conforms with a protocol that has particular strengths and functionality that make the link well suited for use in particular environments. For example, the PCI Express link 114 comprises a high speed, inexpensive short-haul connection, while the Infiniband link 116 employs a high-bandwidth protocol that is useful in data center transmissions. Further, where it is desired to transmit data across an enterprise LAN, the GigE link 118 is quite effective. The SONET link 120 is particularly well adapted for high bandwidth, long haul

applications. Finally, the Fibre Channel link 122 enables data transmission to high performance disk drive storage systems such as the RAID drive storage tower 112.

[0030] As the foregoing suggests, the communications system 100, as well as other operating environments, comprises a variety of different communications links, systems and devices conforming with any number of communications protocols. Such arrangements are useful because they enable users to more fully exploit the relative strengths of the various communications protocols.

[0031] Directing attention now to Figure 2, details are provided concerning a protocol analyzer, or simply “analyzer,” 200 suitable for use in connection with the communications system 100, or other operating environment. As indicated in Figure 2, the protocol analyzer 200 is disposed in an in-line arrangement with respect to each of the components in the communications system 100. In particular, the PCI Express links 114A and 114B enable routing of data from the CPU 102 through the analyzer 200 to the Infiniband adapter 104. As suggested in Figure 2, the Infiniband links 116A and 116B, GigE links 118A and 118B, SONET links 120A and 120B, and Fibre Channel links 122A and 122B likewise enable routing of data through the analyzer 200 and on to the next link in the series.

[0032] Thus arranged, the protocol analyzer 200 receives data traffic from each of the links in the communications system. The illustrated arrangement is exemplary only however and is not intended to limit the scope of the invention. For example, in some implementations, the protocol analyzer 200 receives data from less than all the links in the communications system 100. Moreover, the protocol analyzer 200 need not be positioned in an in-line configuration in every case. Accordingly, in some

implementations, the protocol analyzer 200 is configured and arranged to receive network data from a tap, or taps, on one or more of the links. More generally, the protocol analyzer 200 can be arranged in any way, relative to the communications system 100, that is consistent with the functionality disclosed herein.

## **II. Exemplary Protocol Analyzer Configurations**

[0033] As the foregoing discussion suggests, embodiments of the protocol analyzer may be configured in a variety of different ways. With attention now to Figure 3, details are provided concerning an exemplary link analyzer 300 design configured to implement capture of link data.

[0034] In the illustrated embodiment, the link analyzer 300 includes a serializer/deserializer (“SERDES”) 302 configured to receive and transmit network traffic by way of a communications link (not shown) of a communications system. Generally, the SERDES 302 is synchronized with the transmitted clock on the input link. The link analyzer 300 further includes an analyzer front end 304 and analyzer back end 306.

[0035] As indicated in Figure 3, the analyzer front end 304 is configured to receive a “trigger” signal, such as from another analyzer. Additionally, and as disclosed elsewhere herein, the analyzer front end 304 may also generate and transmit a “trigger” signal in some cases. In similar fashion, the analyzer back end 306 is configured to receive an analyzer clock, which may also be referred to herein as a “reference clock,” such as from another analyzer. As disclosed elsewhere herein, the analyzer back end 306 may also generate and transmit the analyzer clock in some cases.

[0036] Finally, the link analyzer 300 includes a memory 308. Generally, the memory 308 enables the link analyzer 300 to store captured data events and other information and materials that relate to the communications link(s) with which the link analyzer 300 is associated.

[0037] Generally, operation of the link analyzer 300 begins when the link analyzer 300 is set to “arm” and link data begins to flow through the link analyzer 300. The SERDES 302 frames the received link data into data words having a predetermined length. The framing of the data enables the internal logic of the link analyzer 300 to run at a much lower frequency than the link transmission frequency. Ultimately, the SERDES 302 also retransmits the received link data so that the data flow in the link is not interrupted.

[0038] After the SERDES 302 has framed the received link data into data words, the analyzer front end 304 examines link data words received from the SERDES 302 and compares the link data words to one or more specified patterns, or “triggers,” each of which corresponds to a particular error, problem, condition or event in the link data stream and/or in the communications system. In the event that the analyzer front end 304 identifies a trigger within the incoming link data stream, the link analyzer 300 is triggered and the link data words flow from the analyzer front end 304 to the analyzer back end 306. In some implementations, such as in the case of the multi-link protocol analyzer disclosed below in connection with Figure 4, the link analyzer 300 may also be triggered by another link analyzer in the system.

[0039] Directing attention now to Figure 4, and with continuing reference to Figure 3, more particular details are provided concerning an exemplary implementation of a

multi-link protocol analyzer denoted generally at 400 in Figure 4. Generally, the multi-link protocol analyzer 400 serves to monitor multiple communication links and to capture link data for use in system troubleshooting efforts and related evolutions.

[0040] To these ends, the multi-link protocol analyzer 400 includes hardware that is configured to receive and capture data events associated with a particular communication protocol. Such hardware includes one or more pairs of ports, each of which is configured and arranged to interface with a bi-directional communication link. The multiple protocol-specific devices also include hardware and/or software that is adapted to recognize the occurrence of predefined events in the data received by way of the bi-directional communication link.

[0041] As indicated in Figure 4, the multi-link protocol analyzer 400 includes multiple link analyzers, an implementation of which is discussed above in connection with Figure 3. Each of the link analyzers, also referred to sometimes as cards, blades or boxes, is adapted for use with a data stream corresponding to a particular protocol. In particular, each of the link analyzers is configured to capture, process and analyze data from a particular communications link with which that link analyzer is associated. The link analyzers may be modular or interchangeable so as to permit the multi-link protocol analyzer 400 to be readily modified or adapted for use with various types of communications systems.

[0042] In the particular implementation illustrated in Figure 4, the multi-link protocol analyzer 400 includes a first link analyzer 402, a second link analyzer 404 and a third link analyzer 406 arranged in series with each other. Exemplarily, each of the link



analyzers 402, 404 and 406 is configured for use in connection with a different communications protocol.

[0043] The link analyzer 402, for example, is arranged in an in-line configuration so as to receive data from a communications link “1” input, and to pass the received data to a corresponding communications link “1” output. As disclosed in further detail elsewhere herein, the received link “1” data is examined by the link analyzer 402 for the presence of one or more trigger conditions which, if detected by the link analyzer 402, cause the generation and transmission of a trigger signal 402A to the link analyzers 404 and 406. Contemporaneously with generation and transmission of the trigger signal 402A, the analyzer generates and transmits an analyzer, or reference, clock signal 402B.

[0044] As further indicated in Figure 4, the link analyzer 402 is also configured to receive, either directly or indirectly, a trigger signal and analyzer clock signal from the link analyzer 406. The link analyzers 404 and 406 are similarly configured to transmit and receive trigger and analyzer clock signals. Further, the operation of link analyzers 404 and 406 concerning link “2” data and link “3” data, respectively, is analogous to the operation of link analyzer 402 with respect to link “1” data.

[0045] Thus, for example, in the event that the link analyzer 404 detects a trigger condition in the link “2” data, the link analyzer 404 generates and transmits trigger 404A and analyzer clock 404B. In like fashion, if the link analyzer 406 detects a trigger condition in the link “3” data, the link analyzer 406 generates and transmits trigger 406A and analyzer clock 406B.

[0046] It should be noted that while the link analyzers 402, 404 and 406 are shown in Figure 4 as being arranged in serial fashion, the scope of the invention is not so limited.

In some implementations, the link analyzers 402, 404 and 406 are arranged so that a trigger and/or clock signal generated by one link analyzer is propagated in parallel to the other link analyzers in the system.

### **III. Domain Creation and Modification**

[0047] As disclosed herein, multiple virtual protocol analyzers, or domains, can be defined and used in connection with a multi-link protocol analyzer. The definition and use of domains enhances the flexibility of the multi-link protocol analyzer by allowing a user to set up a variety of different experiments directed to obtaining information concerning the performance of the communication links with which the multi-link protocol analyzer is connected. Further, the use of such domains enables users to readily change from one experiment to another, as well as to design and implement new experiments, without necessitating hardware reconfiguration. In at least some cases, multiple experiments may be run simultaneously, so that the resources of the multi-link protocol analyzer can be used to maximum effect and with a high degree of efficiency.

[0048] With attention now to Figures 5A through 5C, details are provided concerning processes for the use of various graphical user interfaces in connection with the creation and modification of domains. Note that while this process is referred to as domain creation, the process is equally well suited to implement changes to existing domains.

#### **A. Port Discovery**

[0049] As indicated in Figure 5A, the first stage 502 in the domain creation process 500 is to identify, or discover, the ports of the multi-link protocol analyzer. The user must be able to ascertain what ports are provided in the multi-link protocol analyzer so that

decisions can be made as to which ports will be included in a particular domain. The identification of ports can be performed in a variety of different ways, based upon the hardware configuration of the particular multi-link protocol analyzer to be employed in the domain creation. For example, where the multi-link protocol analyzer reflects a bus-based design, as in a PCI card, the ports can be identified by scanning the bus. As another example, in networked instruments, a network based discovery mechanism, such as user datagram protocol (“UDP”) broadcasts, can be used to identify the ports of the multi-link protocol analyzer. Any other suitable method of port discovery may be used as well.

[0050] Directing particular attention now to Figure 5B, and with continuing attention to Figure 5A, the process 500 advances to stage 504 after the ports of the multi-link protocol analyzer have been discovered. At stage 504, a list of the discovered ports and information concerning the ports is presented to the user by way of a graphical user interface (“GUI”) 600. In the embodiment of the GUI 600 illustrated in Figure 5B, all of the discovered ports 602 are displayed together in an interleaved fashion, without regard to the respective native protocols of the ports, under a directory 604 entitled “AllDevices.” Of course, the discovered ports may be displayed in various other ways as well.

[0051] More generally, the scope of the invention is not limited to any particular implementation, configuration or arrangement of the GUI 600. Rather, embodiments of the GUI may be configured in any way that is consistent with the functionality disclosed herein. Accordingly, the GUI may include a variety of vehicles configured to display information such as, but not limited to, radio buttons, data fields, directory trees, tables

and charts. Similarly, embodiments of the GUI likewise include a variety of vehicles or mechanisms configured to receive input from a user, wherein such input may take various forms and may be provided by devices such as a mouse or keyboard. Examples of user input received by way of the GUI 600 include, but are not limited to, selection of radio buttons, entry of information into fields provided by the GUI 600, drag-and-drop actions, and select-copy-paste actions.

[0052] For each displayed port, the list includes a description of the port, the associated protocol, and a number that uniquely identifies the port. The identifying number allows the user to relate the listed port to the actual port hardware and can be any type of indicator or information that uniquely describes the associated port. In the illustrated embodiment of the GUI 600, the number assigned to each discovered port corresponds to an associated slot in a bus.

[0053] As further indicated in Figure 5B, the GUI 600 is also configured to present other information concerning the discovered ports. For example, the GUI 600 indicates the communications protocol, such as Infiniband (“IB”), Fibre Channel (“FC”) and Gigabit Ethernet (“GE”), associated with each of the ports. Thus, as indicated in Figure 5B, an exemplary port designation would take the form “FC\_Port-5,” indicating that this port is configured for the Fibre Channel communications protocol and is associated with slot 5 of a bus.

[0054] Another example of information displayed by the GUI 600 concerns the status of each of the ports. Because different ports can be used simultaneously by different users, it is useful for the user to know which ports are available and which are not. To this end, at least some embodiments of the GUI use a color coded icon scheme to

indicate the status of a port. In one example, if the icon 606 is green, the port is available for use. If the icon is blue, the port is in use by another user. Finally, if the icon is red, the port hardware is in an error state and cannot be used. Such a color code scheme is but one example of a way in which port status information can be presented to a user. Any other system or indicator of comparable functionality may likewise be employed.

**B. Port Selection**

[0055] With continuing attention to Figures 5A and 5B, the process 500 advances to stage 506 after the list of the discovered ports and information concerning the ports is presented to the user. Because the user may need to collect data from several communications links at the same time in order to construct an experiment to diagnose a problem, the GUI 600 receives input, at stage 506, from the user concerning the name of a domain to be created or modified, and the ports that are to be included in the domain. In this way, the GUI 600 enables the user to create a virtual multi-port analyzer, or domain, that includes a collection of ports selected by the user. Thus, the user can readily develop a virtual instrument, or multi-port analyzer, tailored to address a specific problem or matter of interest in the communications network. In some cases, the domain thus created or modified is for the exclusive use of the user who has created the domain. However, some embodiments of the invention permit the creator/modifier of the domain to specify a list of users for the domain.

[0056] Moreover, multiple domains can be created at the same time from a collection of non-overlapping ports. From the perspective of the user, a domain appears as a single protocol analyzer in which all of the ports share a trigger line and a common time clock.

The protocol analyzer can establish shared trigger lines and time clocks in any of a variety of ways so long as, from the perspective of the user, the selected ports appear to share trigger lines and a common time clock.

[0057] With particular attention to Figure 5B, the illustrated embodiment of the GUI 600 facilitates domain creation and modification by enabling the user to create and name a domain, such as the “NewDomain” 608 indicated, and to add ports to that domain. Alternatively, the user can remove and/or add ports from/to an existing domain.

[0058] The GUI 600 can be configured in a variety of different way so as to enable the user to add ports to the domain, and/or remove ports from a domain, in a variety of ways. Accordingly, the scope of the invention is not limited to any particular GUI implementation. For example, some embodiments of the GUI 600 include a drag-and-drop feature by which ports are added to, or removed from, the domain. In particular, the user is able to modify the domain simply by dragging ports from the “AllDevices” list to the domain, or dragging ports from the domain to the “AllDevices” list. In one alternative embodiment, the GUI 600 is configured so that the user can use a select-copy-paste technique to add ports to a domain. As the user adds ports to, or removes ports from, a domain, the change in the configuration of the domain is reflected in the display presented by the GUI 600.

[0059] Because domains can be created simply by selecting the ports that are to be included in a desired domain, one or more virtual protocol analyzers can be readily defined and used without necessitating changes to system hardware connections, such as the communications links of the system under test. For example, if a set of links are

initially broken and connected to ports of a multi-link protocol analyzer, embodiments of the invention permit multiple users to select from the set of links to create domains. Thus, once the links have been initially instrumented, any analyzer port can be included any number of times into any domain. Such flexibility allows different users to diagnose different problems using different combinations of ports without frequent physical disruption of the links in the system.

### C. Port Configuration

[0060] Each port in a protocol analyzer may have many different parameters which can be adjusted or changed as necessary to control operational matters such as the type and amount of data to be collected in connection with that port. Accordingly, embodiments of the GUI are configured to enable a user to readily configure, and reconfigure, the ports in a multi-link protocol analyzer. This capability is particularly useful where a single domain includes ports conforming to a variety of different protocols, since multiple port parameter sets must be configured.

[0061] Directing particular attention now to Figure 5C, and with continuing attention to Figure 5A, exemplary embodiments of the GUI 600 enable a user to configure one or more ports of the domain. Thus, after port selection has been completed, the process 500 advances to stage 508 where the user configures one or more ports of the domain. In some alternative implementations, the user is able to configure ports prior to including those ports in the domain. Generally, a configuration editor section of the GUI 600 displays editable operating parameters, as well as fixed port values and characteristics in some cases, for one or more ports and receives input from the user concerning such parameters.

[0062] More particularly, the GUI 600 is configured so that the configuration editor interface 610 changes depending upon the communications protocol supported by the analyzer port. For example, if the user selects a Fibre Channel port, a Fibre Channel configuration editor is displayed as is shown in Figure 5C. Similarly, if the user selects an Infiniband port for example, an Infiniband configuration editor is displayed by the GUI 600. In this way, the user can effectively and efficiently configure the ports in the domain for operation. It should be noted that the GUI 600 may be constructed for use in connection with a wide variety of other communications protocol types as well.

[0063] After the ports of the domain have been configured, the virtual protocol analyzer, or domain, is ready to be used to capture and analyze network data and to display the results. In operation, the ports of the analyzer monitor different links in the system under test. Each port can have different trigger criteria and other settings, including those that have been set using the exemplary graphical user interface illustrated in Figures 5B and 5C. When one port triggers, its trigger is propagated to the other ports in the domain, as illustrated in Figure 2. The ports then capture the network data transmitted on the communications links that interface with the ports. Human-readable data that summarizes the captured data can then be displayed to the user. In general, the methods for discovering the ports of the protocol analyzer and presenting user interfaces that enable the user to create a domain and configure the ports of the domain can be used with any suitable protocol analyzer and with any suitable methods for displaying the results of the protocol analysis.

[0064] Thus, exemplary embodiments of, the invention are concerned with a graphical user interface that, among other things, enables a user to quickly and efficiently create



domains having multiple ports, which may be associated with different protocols, and to configure the ports in the domain. The resulting virtual protocol analyzer appears, from the perspective of the user, to be a dedicated protocol analyzer that shares a trigger lines and a common clock. The domain can be reconfigured by the same or different users as needed to prepare the domain for use with different sets of communications links in the system under test.

#### **IV. Computing Environments, Hardware and Software**

[0065] In at least some cases, some or all of the functionality disclosed herein may be implemented in connection with various combinations of computer hardware and software. For example, at least some protocol analyzers use hard coded devices such as field programmable gate arrays (“FPGA”) to implement timestamping, data sorting and data capture functionality. Other protocol analyzers employ both hardware and software to implement various functions disclosed herein.

[0066] With respect to computing environments and related components, at least some embodiments of the present invention may be implemented in connection with a special purpose or general purpose computer that is adapted for use in connection with communications systems. Embodiments within the scope of the present invention also include computer-readable media for carrying or having computer-executable instructions or electronic content structures stored thereon, and these terms are defined to extend to any such media or instructions that are used with telecommunications devices.

[0067] By way of example such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to carry or store desired program code in the form of computer-executable instructions or electronic content structures and which can be accessed by a general purpose or special purpose computer, or other computing device.

[0068] When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer or computing device, the computer or computing device properly views the connection as a computer-readable medium. Thus, any such a connection is properly termed a computer-readable medium. Combinations of the above should also be included within the scope of computer-readable media. Computer-executable instructions comprise, for example, instructions and content which cause a general purpose computer, special purpose computer, special purpose processing device such as a link analyzer or multi-link protocol analyzer, or computing device, to perform a certain function or group of functions.

[0069] Although not required, aspects of the invention have been described herein in the general context of computer-executable instructions, such as program modules, being executed by computers in network environments. Generally, program modules include routines, programs, objects, components, and content structures that perform particular tasks or implement particular abstract content types. Computer-executable instructions, associated content structures, and program modules represent examples of program code for executing aspects of the methods disclosed herein.

[0070] The described embodiments are to be considered in all respects only as exemplary and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

WORKMAN NYDEGGER  
A PROFESSIONAL CORPORATION  
ATTORNEYS AT LAW  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84111